Text detection in street level images

Jonathan Fabrizio, Beatriz Marcotegui, Matthieu Cord

Pattern Analysis and Applications manuscript No.
(will be inserted by the editor)

Text detection in street level images

Jonathan Fabrizio, Beatriz Marcotegui and Matthieu Cord

Abstract Text detection system for natural images is a very challenging task in Computer Vision. Image acquisition introduces distortion in terms of perspective, blurring, illumination, and characters may have very different shape, size, and color.

We introduce in this article a full text detection scheme. Our architecture is based on a new process to combine a hypothesis generation step to get potential boxes of text and a hypothesis validation step to filter false detections.

The hypothesis generation process relies on a new efficient segmentation method based on a morphological operator. Regions are then filtered and classified using shape descriptors based on Fourier, Pseudo Zernike moments and an original polar descriptor, which is invariant to rotation. Classification process relies on three SVM classifiers combined in a late fusion scheme. Detected characters are finally grouped to generate our text box hypotheses. Validation step is based on a global SVM classification of the box content using dedicated descriptors adapted from the HOG approach.

Results on the well-known ICDAR database are reported showing that our method is competitive . Evaluation protocol and metrics are deeply discussed and results on a very challenging street-level database are also proposed.

J. Fabrizio
LRDE-EPITA lab., 14-16, rue Voltaire, F-94276 Le Kremlin Bicetre cedex, France
E-mail: jonathan.fabrizio@lrde.epita.fr

B. Marcotegui
MINES ParisTech, CMM lab., 35 rue Saint Honore 77305 Fontainebleau cedex France
E-mail: marcoteg@cmm.ensmp.fr

M. Cord
UPMC-Sorbonne Universités, LIP6 lab., 4, place Jussieu, 75005 Paris, France E-mail: matthieu.cord@lip6.fr

Fig. 1 Images from ITOWNS project.

Keywords Text detection · Text segmentation · TMMS · Toggle mapping · image classification

1 Introduction

Text detection in images is a major task in computer vision. Applications are various: content-based image indexing [30], document image analysis [54], visual impaired people assistance [39,18,10]. Generic text de-

tection is a very difficult problem and most works investigate specific contexts: postal address box localization [37], license plate localization [2], text extraction in video sequences [52], automatic forms reading [27] and globally *document* oriented systems [51]. Such contexts constraint the problem and many hypotheses are assumed (size, position, alignments, temporal redundancy in video sequences...). In natural images, as street level images shown in Fig. 1, it is difficult to make realistic hypotheses on text style, color, *etc.*. Text detection is then a much more difficult task and most of the previous methods do not apply without considerable changes.

In this paper, we introduce a new approach for text localization in natural images. Regarding the methodology, our approach relies on the hypothesis generation/validation paradigm: our process is carried out using an efficient segmentation algorithm, followed by classification and grouping processes. Then, the hypothesis validation step operates with texture-based descriptors computed over text regions.

Related work is presented in the next section and our contributions and innovations regarding previous works are introduced. Our general methodology is also explained, and all steps of the algorithm are detailed in sections 3, 4 and 5. Results on real street view images are exposed in section 6 and the system efficiency is evaluated on ICDAR database in section 7.

2 Related work

The state of the art [34, 26, 29] usually splits methods into two groups:

- The so-called *connected component* approaches, looking for characters by segmenting images, selecting relevant regions, and then looking for text from the detected regions. For instance in [41], the image is segmented using a morphological operator (ultimate opening). The segmented regions are then classified as text or not, and a grouping step is applied in order to obtain large text regions. We can extend this type of approaches to all methods that extract local regions (or points, lines, ...) of interest in the image, corresponding to character or letter candidates before the grouping step. For instance, methods proposed in [14] [32] are based on edge detection, and then morphological or smoothing filtering are applied to connect and merge regions. Unfortunately, these methods are not robust enough when the background has similar strong edge distributions as the text regions..
  Character segmentation has been extensively studied in document analysis context. Most of the existing approaches are unusable in natural scenes as they assume to include a global background, such as the method of Seeger and Dance [44] and the classical Otsu criterion [36]. In generic context, local thresholding and morphological filtering stand out from the literature.
- The so-called *texture-based* approaches, straightly localizing text boxes into images by analyzing general characteristics of text over windows. Most of them are inspired from the works of Viola and Jones [50] on object detection by cascade of weak classifiers. This scheme has been adapted for license plate detection [2] and by Chen and Yuille in [7] for text detection, substituting Haar wavelets with X and Y derivatives. The main advantages of such approaches are speed and simplicity. However, the descriptors used are usually not robust to geometric transformations, thus leading to use huge training data set.

To evaluate performances, 2 main competitions (ICDAR 2003 and 2005) have been organized. Published results and methods stay very competitive until now. The evaluation protocol is very strict. Several recent papers as [38] report very nice results but not strictly using the same protocol as ICDAR. So, they are not really comparable. Note that very few other competitions have been done recently [21].

Connected component approaches have proved to be more efficient for this task than texture-based approaches. ICDAR text locating contest [33] and ImagEval evaluation campaign [21] ranked a connected component approach at the first place [41]. Another evidence of *Connected component* effectiveness is their recent use for text detection in video context [55].

This analysis motivates our strategy to develop a hybrid system combining both approaches. We first propose a detection stage based on a *connected component* strategy, and we add a *texture-based* validation stage to filter out false detections.

Our proposition is a bottom-up process for the hypothesis generation looking first for characters, and secondly for words. It is very different from [6] that proposes a top-down strategy for hypothesis generation and a validation scheme based on a *connected component* method.

As far as we know, no system has been proposed using such combination for text detection. We published a preliminary work on text detection system [11] that only proposes a basic implementation of the *connected component* strategy. The full system proposed in this paper includes a validation strategy, that is definitively a deep change and a substantial contribution. Quantitative evaluation is also missing in [11]. Databases, pro-

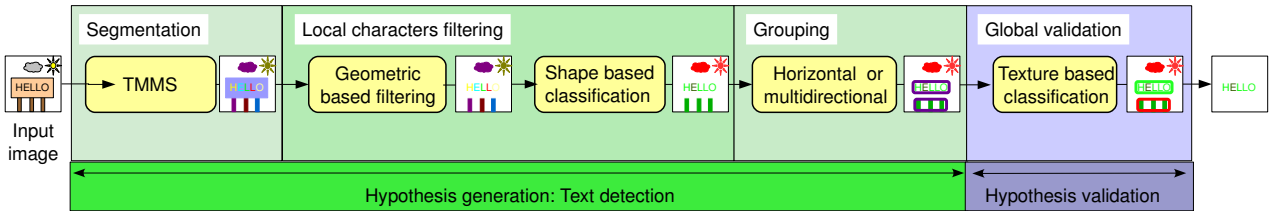


Fig. 2 Full scheme of our text detection process. Starting with an input image, segmentation, geometrical and shape-based filtering are applied in order to induce character candidates for a grouping process. At the end of this first step, text box hypotheses are proposed to the final validation process which is based on a box classification.

tocols and accurate quantitative and qualitative evaluations on databases are proposed in this paper. This is a significant improvement compared to the previous publication.

In Fig 2, the whole process is reported:

1. Hypothesis generation: the first block, TMMS (Toggle Mapping Morphological Segmentation), is the image segmentation process. To achieve this step, we introduce a method based on the *toggle mapping*, a morphological operator. Regions are then filtered and a binary classifier induces character and non-character labels. The regions classified as characters are grouped altogether with other surrounding character regions to form regions containing words, sentences, or text blocks.
2. Validation step: the aim of this step is to validate text boxes while removing false positives. For this purpose, a classifier is trained using texture-features computed on the detected candidate text boxes.

Details about the implementation and original contributions for each block of our system are given in the following sections.

3 Segmentation process

3.1 Introduction

A wide variety of criteria already exists to binarize an image [46,5]. In our context, a local criterion is required. Two local criteria have been widely used in the literature for character segmentation: Niblack criterion [35] and Sauvola criterion [43,42]. The Niblack threshold $T(x)$ for a given pixel x, according to its neighborhood, is:

$$T(x) = m(x) + ks(x) \tag{1}$$

with m and s the mean and the standard deviation computed on the neighborhood and $k \in \mathbf{R}$ a parameter. The Sauvola binarization criterion [43] evaluates a threshold $T(x)$ by:

$$T(x) = m(x)\left(1 + k\left(\frac{s(x)}{R} - 1\right)\right) \tag{2}$$

with R the dynamic of standard deviation $s(x)$.

Shafait et al. in [47] speed up computation of the mean and standard deviation (in a constant time regardless of the neighborhood size), using *Integral Image*. Notice that recently a variation of Niblack has been exposed by Kaihua et al. in [56]: the NLNiblack (Non-Linear Niblack).

These two criteria are efficient in document like images but their performances decrease on natural images [13]. Another approach based on boundaries, defines a stroke filter [31]. This filter is an edge detector designed for character stroke and is used for text detection [25].

Morphological approaches are also used for this segmentation step. Introduced by Beucher in 2005, the Ultimate Opening (UO) is a residual operator that highlights patterns with the highest contrast [3]. The operator successively applies increasing openings and selects the maximum residues computed between two successive results of opening. This operator is efficient and does not need any parameter. It has been used before in text localization by Retornaz [41] and this system was ranked first during imagEval evaluation campaign in 2006 [21]. The implementation of this operator is time consuming, but a real time implementation using a maxtree representation has been recently proposed [12].

Morphological framework is definitively relevant for our segmentation task. We introduce in the following a fast and efficient method based on a morphological operator introduced by Serra: *Toggle Mapping* [45].

3.2 TMMS segmentation method

Toggle mapping is a generic operator which maps a function on a set of n functions: given a function f (defined on D_f) and a set of n functions $h_1, ..., h_n$, this operator defines a new function k by assigning to each

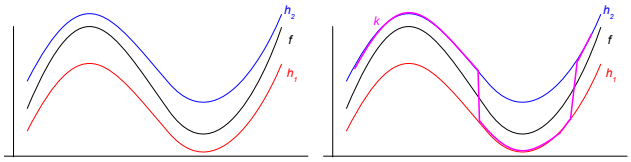


Fig. 3 On the left, function f and a set of 2 functions h_1 and h_2. On the right, function k computed by toggle mapping.

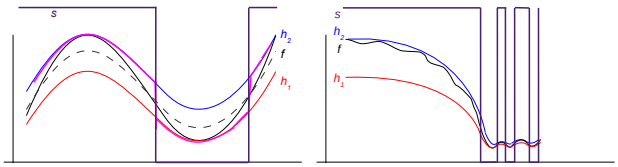


Fig. 4 Result of Eq. 6 (function s) on an edge and in homogeneous noisy regions.

pixel x the closest function h_i among the set $(h_1, ..., h_n)$ (Fig. 3):

$$\forall x \in D_f \; k(x) = h_i(x); \forall j \in \{1..n\} \\ |f(x) - h_i(x)| \leq |f(x) - h_j(x)| \qquad (3)$$

A classical use of toggle mapping is contrast enhancement: this is achieved by applying toggle mapping on an initial function f (an image) and a set of 2 functions h_1 and h_2 anti-extensive[1] and extensive[2] respectively, usually morphological dilation and erosion.

These two functions are computed by:

$$\forall x \in D_f \quad h_1(x) = \min f(y); y \in v(x) \qquad (4)$$

$$\forall x \in D_f \quad h_2(x) = \max f(y); y \in v(x) \qquad (5)$$

with $v(x)$ a given neighborhood (the structuring element) of pixel x.

We propose here to use Toggle mapping operator for segmentation. Instead of considering a function k (Eq. 3), we just keep the index of the function on which the pixel is mapped. This leads us to define a function s as:

$$\forall x \in D_f \quad s(x) = \underset{i \in \{1,2\}}{argmin} |f(x) - h_i(x)| \qquad (6)$$

Two additional parameters are introduced: (1) a minimal contrast c_{min}. In highly contrasted regions, our segmentation is robust against small local variations. However, in homogeneous regions, the difference between mapping functions and the original one is usually very low. In that case, function s (eq. 6) frequently switches from one to another, leading to a salt and pepper noise (see Fig. 4 right). Including a parameter c_{min} (see equation 7) avoids this problem. (2) a parameter $p \in [0, 1]$ to manage the thickness of the detected structures[3]:

$$s(x) = \begin{cases} 0 \; if |h_1(x) - h_2(x)| < c_{min} \\ 1 \; if |h_1(x) - h_2(x)| >= c_{min} \\ \quad \& \; |h_1(x) - f(x)| < p * |h_2(x) - f(x)| \\ 2 \; otherwise \end{cases} \qquad (7)$$

[1] f anti-extensive $\Leftrightarrow f(X) \subset X$

[2] f extensive $\Leftrightarrow X \subset f(X)$

[3] The smaller p, the more probably pixels are assigned to the high value areas (Fig. 5).

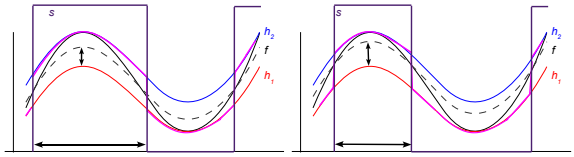
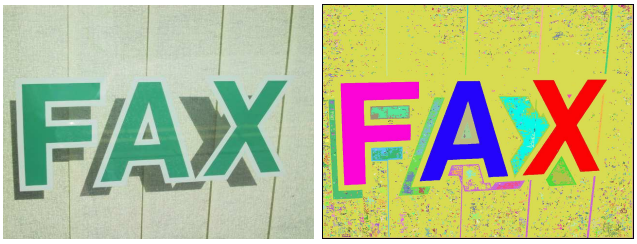


Fig. 5 Influence of parameter p on thickness: variations of p change the position of the boundary decision.



Original image. Segmented image.

Fig. 6 TMMS segmentation result on the FAX image (from ICDAR database).

Fig. 7 Results of the TMMS segmentation on street-level images.

Finally, very small homogeneous regions are merged into their bounding ones and all regions are labeled to provide the segmented image. This method is called Toggle Mapping Morphological Segmentation (TMMS).

Results are shown in Fig. 6 and Fig. 7. p is empirically set to 0.8 with a 7x7 or 9x9 square structuring

element. Extending [13], we use hysteresis to robustly set up c_{min}. These values are used for all our experiments. In order to ensure the effect of c_{min} perceptually uniform, the threshold should be applied to the gamma corrected image, if the gamma value of the acquisition system is known.

We have also implemented several recent segmentation methods discussed in section 3. We reported many comparisons in our previous work [13] showing that TMMS is very fast, provides good segmentations of characters, less noisy results with far fewer regions than other algorithms, increasing the segmentation quality. Note also that our TMMS algorithm was ranked second out of 43 methods in DIBCO (Document Image Binarization Contest) challenge of ICDAR 2009 [15]. We have kept our method in our scheme instead of using the ranked first method because, according to table 3 in [16], this method is better on handwritten text but our method outperforms it on machine printed text which is more our goal. Moreover, this method is well adapted for documents but not for natural images. Especially the first step of the method is designed to remove background and is not directly usable on our photos.

4 Character Detection and Grouping

4.1 Geometrical Filtering

After segmentation, the system must be able to select regions which represent characters. Some of these regions are obviously not text and may be identified with simple criteria (i.e. too large/too small...). These regions are quickly detected and removed without removing any (*as less as possible*) character. These fast filters help to save time during the next steps. The strategy acts in 3 steps:

- First, a collection of fast filters (attribute opening [4]) based on simple attributes (width, height and surface) is applied,
- Then, as Retornaz does in [41], the density of regions is estimated. Regions in high density areas are removed. The density is estimated by dilating all regions in the image and by counting the number of regions connected by this dilation (Fig. 8). Most of the time it removes complex textured zones, such as trees, and then removes a lot of possible false positives.
- Finally, isolated regions are removed, making the assumption that characters are not alone.

Figure 9 shows the simplification performed by fast filters applied on the image of figure 6.

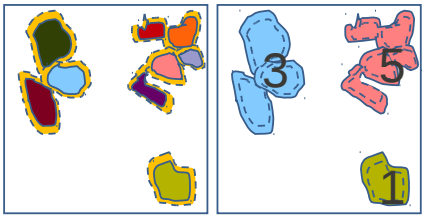


Fig. 8 The density of regions is estimated by dilating all regions and counting merged regions.

Fig. 9 Geometrical filter result on the FAX segmented image. Black regions are dismissed, white regions are letter candidates for classification step.

4.2 Shape Classification

Previous filters remove a large part of regions which are not characters, but to go further, a binary classification with suitable descriptors is proposed.

Due to the variability of the analyzed regions, it is interesting to get shape feature representation invariant to rotation and scale transformations. Additionally, the training database may help to be more robust to perspective deformations. We selected Fourier descriptors and pseudo Zernike moments that have demonstrated their performances for character recognition [49].

We add a third representation based on a polar analysis [48]. In polar coordinate space, centered onto its gravity center, the shape is mapped into a normalized rectangle (Figure 10). The representation is then invariant to scale factor. To provide rotation invariance, a vertical histogram (horizontal projection) is commonly used. However, much information on the character shape is then lost. Instead, Szumilas *et al* [48] redefine the distance computed between samples, keeping full polar description, but requiring a very high complexity computation.

We propose in this article a new rotation invariant descriptor: a rotation of the shape is seen as an horizontal translation in the normalized rectangle (in polar coordinates representation - Figure 10). To be robust to this translation we take the spectrum magnitude of

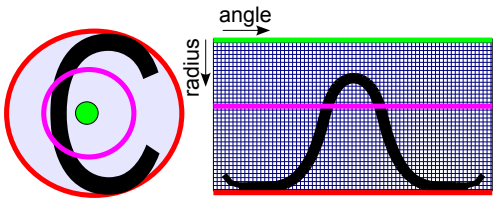


Fig. 10 The region is expressed in a polar coordinate space and to have a rotation invariant descriptor we take the spectrum of Fourier transform of every line.

Fig. 11 Samples of learning database.

the Fourier transform of each line in the polar representation. The descriptor is made by the concatenation of all resulting rows. This strategy encodes much more information than simple histograms, and is much simpler than redesigning distances while preserving rotation invariance property.

We use SVM [8] with RBF kernels to classify the features. However, many combinations of descriptors are possible in the classification framework [28]. In the following, we investigate different combinations that we empirically evaluate to select our final strategy.

We build a training database containing 16200 samples of characters (different sources such as imagEval campaign [21], and ITOWNS images have been used) and 16200 negative samples (Fig. 11). The testing database is composed of 1800 samples of characters and 1800 samples of other patterns. All trainings were performed by the tools provided by Joachims [24,23].

Let us summarize the different empirical optimization of our classification framework:

- SVM classifier with pseudo-Zernike moments (with the 6 first degrees) gives 83.36% of good classification rate, SVM classifier with Fourier descriptors (with outline length of the shape normalized to 30 units) 89.50% and SVM classifier with our polar descriptors (in a 30x30 matrix) 88.67%. These scores can be improved by taking higher degrees of moments but there is a trade-off between the classification rate and speed.
- Different early and late fusion schemes were tested and the late fusion scheme gives the best results. In particular, the early fusion scheme with only one SVM with all descriptors leads to a worse classification rate. We were inspired by boosting techniques to decompose each SVM classifier in the first stage by multiple SVM classifiers but this does not give good performances. For example, learning 9 SVM classifiers on Fourier moments (each one with a subset of the training database) does not lead to 88% classification rate, whatever combination strategy is used.
- In late fusion, multiple combination strategies were tested to take the final decision and the best one was to use another SVM classifier that takes the result of previous SVM classifiers as input. This last SVM classifier reaches the best performances (92.89%).

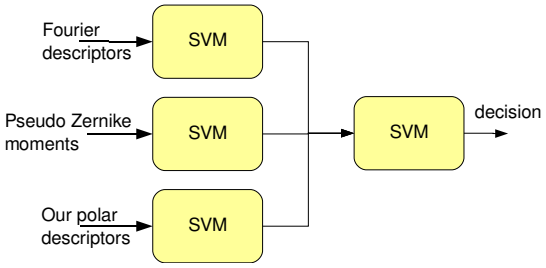


Fig. 12 Our character classification scheme: 3 SVM classifiers working on our 3 descriptors and a late fusion based on a SVM classifier.

Fig. 13 Result of our classification process on the FAX image: regions from figure 9 are classified as text (green regions) or not text (red regions).

Figure 12 summarizes our final strategy: a late fusion scheme with first three SVM classifiers for each family of descriptors, and their output scores recombined into one input vector for a new SVM classifier.

Figure 13 shows the classification result on the FAX image..

4.3 Grouping

The last step consists in a simple grouping procedure of the detected regions to form words, lines, or text blocks. According to figure 14, we note H_1 and H_2 the height of the two regions, ΔC_y the distance of the two

gravity centers along the y-axis and ΔX the distance between the two bounding boxes along the x-axis. Three conditions are tested before linking two regions:

- height similarity: $\frac{|H_1-H_2|}{min(H_1,H_2)} < S_1$
- alignment: $\frac{\Delta C_y}{min(H_1,H_2)} < S_2$
- vicinity: $\frac{\Delta X}{min(H_1,H_2)} < S_3$

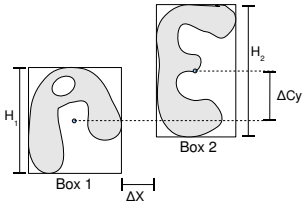


Fig. 14 Measures used in grouping step.

In all our experiments, $S_1 = 1.1$, $S_2 = 0.7$ and $S_3 = 1$ as in [40], and text boxes with less than 2 characters are removed. Figure 15 shows a text detection result.

Fig. 15 Result of the grouping process: a bounding box is surrounding the detected text region.

5 Hypothesis validation

Our hypothesis generation process aims at detecting as many text areas as possible. However, due to periodical pattern or structured features, false positives may be detected. Successive barrels or windows may be seen as a collection of I. Successive circles may be seen as a collection of O. Such situations are very common. Even more, it is possible to find a lot of patterns that look like characters in complicated textures. A classical example is the picture of a tree, branch and leaf generate multiple regions and many of them look like letters. According to our strategy, then our scheme will not surprisingly find text in periodic patterns and textured areas like in trees. At the level of connected component, it is difficult, in these situations, to take the correct decision, this is the reason why we decide to keep a permissive detection as hypothesis generation: our hypothesis generation process aims at detecting as much text areas as possible.

To filter out false positives, a common way is to use O.C.R. with a dictionary to dismiss unlikely text. This should work on many false positives but the use of a dictionary restricts the detection to a specific context. Moreover, in urban scenes, it is impossible to collect all possible words.

Instead we propose a texture-based hypothesis validation step. Most texture-based approaches use features computed on sliding windows at different scales because the text size is not known *a priori*. In our case, given that the candidate text box is known, the process becomes much simpler: a single window is used (instead of sliding windows) at a given scale.

The main feature used to validate regions is the Histogram of Oriented Gradients (HOG) [9]. This histogram of gradients is computed for 10 directions on the whole box (instead of dividing the window in sections as proposed by Dalal).

As we have computed internally a lot of descriptors over grouped characters, we propose to take advantage of these data. To improve the validation decision we concatenate our 10 dimension HOG descriptor with the following descriptors:

- the number of letters detected in the box,
- the surface mean of the letters,
- the perimeter mean of the letters,
- the thickness mean of the regions,
- the mean and standard deviation of surface over the convex hull surface ratio.

The number of letters in the box is provided by the grouping process. The surfaces and perimeters of regions in a box are given by a simple labeling process after the segmentation step. These values do not require additional computation as they are already computed. The thickness of letters is coarsely estimated by the use of an approximative skeleton of letters estimated by successive erosions. The last descriptor is related to the concavity of letters in the potential text box. It involves computing the convex hull of each letter in the text box.

As for detection step, we use a SVM classifier for this validation step on our global features. Linear, polynomial, triangular and RBF kernels are tested. Best performances are reached with a RBF kernel. SVM parameters (kernel parameters, soft margin parameter...) have been set empirically.

To be efficient, such a validation process must not be redundant with the localization process: our localization process generates hypotheses by using local shape-based features and our validation process takes the de-

Fig. 16 Illustration of texture-based validation process: validated hypothesis are blue boxes, rejected hypothesis are red boxes.

cision on global texture-based features, over the complete text box. Such texture-based approaches are not efficient enough for text detection tasks compared to connected component approaches. However, connected component-based detector cannot infer correctly in some situations as they do not consider context (as explained before) while texture-based can. The combination of the two approaches takes advantage of both strategies as more text will be detected by the connected component approach and ambiguous cases will be solved by the texture-based validation by the usage of surrounding information.

Fig. 16 illustrates the validation process: the two text box candidates in Figure 15 are checked and the texture-based validation process rejects one of them (red line) and accepts the other (green line).

This validation step significantly improves the performance of the system, increasing the precision with a slight reduction of recall.

6 Real street view data experiments

Results on real street level images are presented in this section. Images are provided by IGN [4] in the framework of ITOWNS project[5]. A moving vehicle, equipped with capture devices, takes simultaneously a set of 10 images every 4 meters (Fig. 17 - this panoramic image is computed by the combination of one set of images).

In this context, our text detection system has multiple uses: it may help the user to navigate in the image flow by highlighting text boxes, it facilitates image indexing by extracting text information and possibly road sign text. Other applications of the system are planned such as automatic license plate blurring to protect the private life of people in the image flow.

One original picture and intermediate results are reported on figures 18, 19, 20, and 21. The final result of our text detection process is given on figure 22. One can see that most of the text areas have been detected.

Figure 24 shows several examples of text detected by our system. Note that our algorithm applies to real street complete images, such as those of figures 22, 18 or 1. Resulting images of figure 24 are cropped so that text zones can be seen correctly. We can observe that very heterogeneous text zones are detected, with different colors, font styles and orientations proving our system robustness.

Figure 23 illustrates the performance of the validation step. Red boxes are rejected text zones, considered as false positives, while blue ones are validated text zones. We can observe that all rejected candidates (red boxes) correspond to false positives such as window alignments, barriers or tree texture, proving the suitability of the validation step.

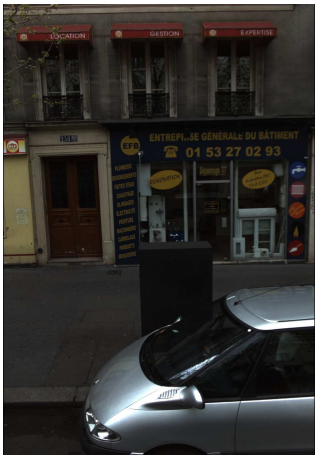


Fig. 18 Original image.

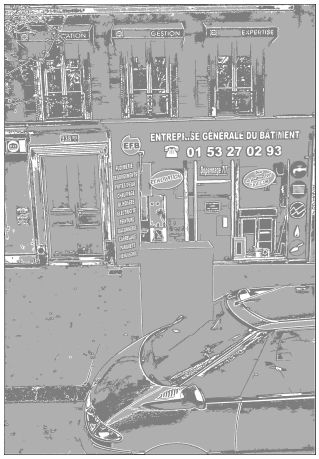


Fig. 19 Segmented image.

There is no available ground truth on this dataset, but we manually evaluate few images, and the results are good for the application. Finally, our system gives similar results on about 3000 images, with fixed parameters and no additional training.

7 Global System performances

Quality assessment protocol is first introduced. Experiments are then reported and discussed.

[4] French national geographic institute IGN [20].

[5] Image-based Town On-live Web Navigation and Search engine [22], a project funded by the ANR (French National Research Agency [1]) and the french consortium Cap Digital. The first goal is to allow users to navigate freely within the image flow of a city and the second is to automatically enhance cartographic databases by extracting features from this image flow.

Fig. 17 Panoramic image from ITOWNS project.

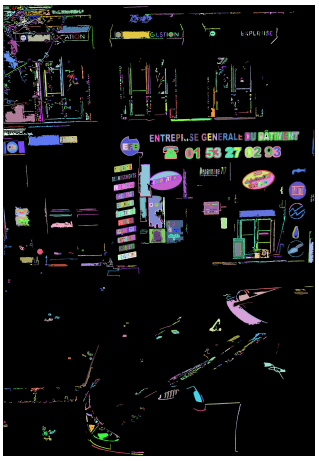


Fig. 20 Geometric filtering result of segmented image in Fig 19.

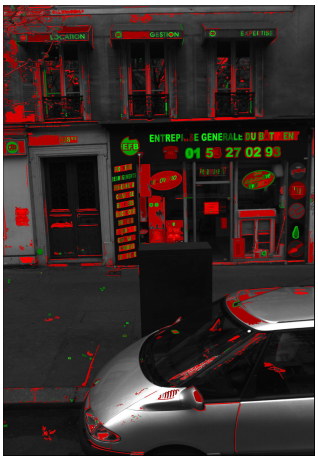


Fig. 21 Letter classification result (letter in green , non letter in red). The lighter the color, the more confident the classifier.

Fig. 22 ITOWNS image result of our text detection. All the boxes of detected texts have been superimposed in green color.

7.1 Evaluation protocol

To evaluate and compare the efficiency of our process with state-of-the-art approaches, the ICDAR text localization competition [19,33] dataset and evaluation protocol are used. This database is provided with a ground truth marking bounding boxes including text. The quality of the detection is evaluated through precision and recall defined as $p = \frac{\sum_{r_e \in E} m(r_e, T)}{|E|}$ and $r = \frac{\sum_{r_t \in T} m(r_t, E)}{|T|}$ where $m(r, R)$ defines the best match for a rectangle r in a set of rectangles R. T and E are the sets of ground-truths and estimated rectangles respectively.

In order to get relevant evaluation, the first difficulty comes from the lack of consistency of the ground truth. Fig. 25 illustrates that the context (logo...) sometimes helps to recognize *unreadable* text (such as partially masked text for example). Conversely, some texts (as reflexion of the text on reflective surfaces or tags are systematically missed by human (Fig. 26)). These artifacts penalize algorithms[6].

[6] A solution would be to do harsh annotation with three classes instead of two: one for the text, one for non-text and the last one for unreadable text. The system is not penalized, whether it detects unreadable text or not.

The second observation deals with the granularity of the annotation, *i.e.* characters, words or sentences granularity level. For example, if the ground truth uses boxes of words, any algorithm providing bounding boxes of sentences is harshly penalized, even if its detections are correct..

Another observation is that, for a group of images, precision and recall are computed for each image, and then averaged over all images. Thus, the weight of a text box is not the same, depending on the number of text boxes in the image.

ICDAR protocol computes precision and recall after a one-to-one matching process between the ground

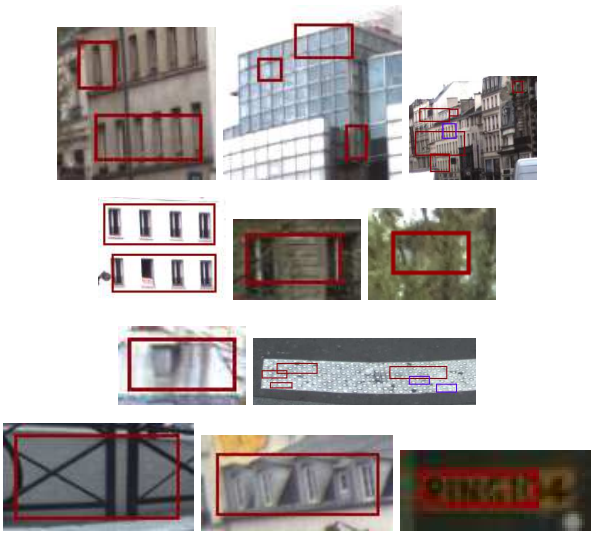


Fig. 23 Examples of detection boxes rejected by our validation process.

truth and the detected bounding boxes[7]. This protocol leads to unsatisfactory results when there are granularity differences between the ground truth and the detection algorithm. Wolf and Jolion in [53] offer another metric which allows one-to-many and many-to-one matching but results are still penalized. For our tests, all scores are given with both ICDAR metric and Wolf metric.

7.2 Results

We evaluate our algorithm on the *TrialTest* part of ICDAR database [8], and we report ICDAR 2005 best results:

	ICDAR metric	Wolf Metric
Our run	$\mathbf{p = 0.63}$ $r = 0.50$	$p = 0.67$ $r = 0.61$
ICDAR 1st [33] Hinnerk / Becker (not published)	$p = 0.62$ $r = 0.67$	– –
ICDAR 2nd [33] Alex Chen	$p = 0.60$ $r = 0.60$	– –
ICDAR 3rd [33] Ashida	$p = 0.55$ $r = 0.46$	– –

Our system leads to performances among the best reported results on this database. Figures 27 to 34 present results.

[7] Following the protocol is important. Using ICDAR database, but changing the protocol can have a significant impact on the performance evaluation.

[8] Our run gets results for the original and sub-sampled images to catch all text data sizes.

Fig. 24 Examples of text boxes from many different ITOWNS images detected by our system.

Figure 27 illustrates several successful results on various ICDAR images. Moreover, our system was developed in the framework of itowns project, and the program must not be tuned to work with different image databases. This is a strong point of our system, proving its robustness to generic image databases in an open context.

On ICDAR, the computation time is on average 15 sec per image. However our code is a research prototype which can be greatly optimized.

Validation step contribution

The hypothesis generation step correctly localizes many text zones. However, as expected, a lot of false positive appear. Thanks to the hypothesis generation and validation scheme, many false positives are removed. Figure 28 shows many results from which approximatively 50% of false positives are removed (red boxes in

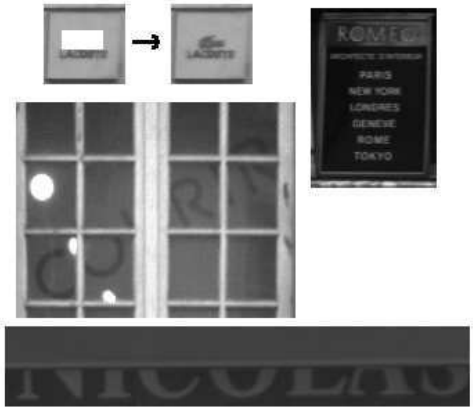


Fig. 25 Annotation issue: should these texts be annotated or not ? Various tests have proved that the first text (Lacoste) is understood by the reader by the use of the logo above (the crocodile). The list of cities (Paris - New York - Londres - Geneve - Rome - Tokyo), on Romeo sign, is readable as soon as you understand it is a list of cities. Partially masked text is also a problem: are the word *courir* (*to run* in french) and the name *Nicolas* they really readable ?

Fig. 26 Annotation issue: should these texts be annotated or not ? This text is readable but most of the time this text is irrelevant: Tags (GPS) and reflexion (Lacoste) or hard to see for a human being (Taxis) ?

the figure). Various kinds of false positives are detected: both periodic patterns (like for the two first images with windows and grids) and textured patterns (like for tree, grass and other for the other pictures).

We have quantified the benefit of the validation step on ICDAR database: precision is 5% better without reducing the recall. The validation step rarely removes correct text boxes: for the whole test part of ICDAR database, this occurs only once (Fig. 31). Our goal is reached as we set the SVM bias to be restrictive in order to avoid removing correct text boxes.. These results validate our hypothesis-validation scheme: the connected component strategy is efficient to detect most text areas (true positive), and the texture-based validation is

Fig. 27 Various results on ICDAR database. The detection succeeds in various contexts: uncommon typeface, various color and textured background.

able to remove most wrong detections (false positive) using a more global context.

Usually, detection fails on isolated characters (not considered in our process) and on low contrasted images (Fig. 29). Periodic structures produce many of the false positive hypotheses (Fig 30), that are filtered in the validation step. All results on other images show that our scheme is robust to various text styles.

7.3 Further discussion on ICDAR benchmark

We illustrate in this section the issues with both the ground truth and evaluation protocol introduced in section 7.1.

Ground truth limitation

The test part (*TrialTest*) of ICDAR database contains duplicates (around 15 images over the 250 ones - with differences in ground truth). 2 images do not have ground truth annotations and some contain mistakes (we have listed more than 25 images with wrong annotations). For example, in images (figure 32) the word “available” and the word “first” at the bottom right corner are missing in the ground truth. Our detection correctly

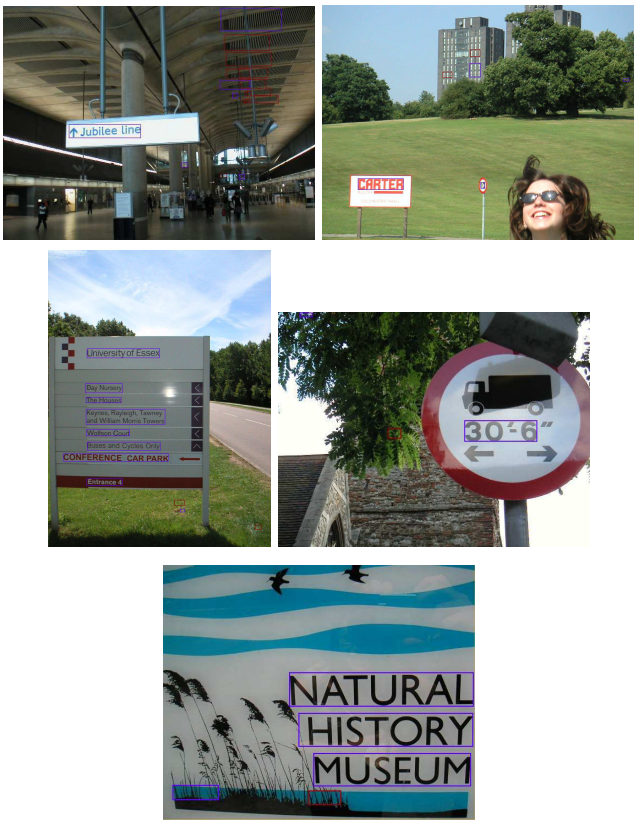


Fig. 28 The hypothesis generation and validation scheme removes a lot of false positives (red boxes).

Fig. 29 Various errors in text detection: isolated characters (letter D), not enough contrasted text (written in the black part of the right image).

handles text in these two images but according to the following table, which gives scores on these images, the evaluation is not representative for them:

	ICDAR metric	Wolf Metric
image 1	$r = 0.52$ $p = 0.65$	$r = 0.83$ $p = 0.87$
image 2	$r = 0.94$ $p = 0.63$	$r = 1$ $p = 0.67$

Some annotations are questionable in Fig. 32, the word "entrance" is visible in the shadow of the sign "Zone ENDS". This word is hard to read but is present, should it be marked or not ? The question is even more conceptual in Fig 33: is the cross a letter or not ?

Fig. 30 Periodic structures frequently cause false positives (consecutive windows and barriers).

Fig. 31 The only image where the validation step fails: The word *SAC* is removed from the detection.

Fig. 32 Some results on ICDAR database on wrongly annotated images. In the ground truth, the word "available" and the word "first" are not marked (Also in the shadow of the sign, the word "Entrance" is not marked in the ground truth but this word is hardly readable).

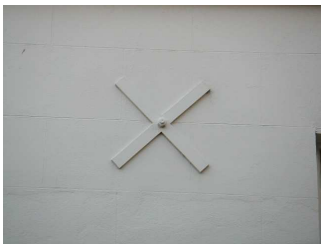


Fig. 33 An image of a feature marked as text in ICDAR ground truth.

Metrics

Evaluation metrics may also skew the results. Figure 34 shows results on ICDAR database where the text is correctly handled by our algorithm. However, very low p and r scores are obtained on theses images:

Fig. 34 Some ICDAR results of our method: even if the text detection is quite good, p and r statistics are very poor on these images because of ICDAR ground truth.

	ICDAR metric	Wolf Metric
image 1	$r = 0.43$ $p = 0.48$	$r = 0.8$ $p = 0.8$
image 2	$r = 0.56$ $p = 0.76$	$r = 0.8$ $p = 0.8$
image 3	$r = 0.39$ $p = 0.43$	$r = 0.8$ $p = 0.8$
image 4	$r = 0.52$ $p = 0.65$	$r = 0.8$ $p = 0.9$
image 5	$r = 0.48$ $p = 0.95$	$r = 0.8$ $p = 0.8$

The first point to explain these unexpected low values it to say that the score is weighted by the number of text boxes in the image. Second, the score is strongly penalized when the granularity of the detection does not match the granularity of the ground truth. Our detection is sentence level detection, the ground truth is word level annotation, this means that in many cases, with ICDAR metric, only one word of a sentence is considered as correct, and the others are considered as false positives. With Wolf Metric, all words from one sentence may have been accepted but the score is penalized as boxes have been broken or merged.

Fig. 35 Results on ICDAR image.

Finally, let us mention a last problem coming from the computation of p and r. Figure 35 shows a well detected text using our method. The following table gives the associated score:

	ICDAR metric	Wolf Metric
image 1	$r = 0.93$ $p = 1$	$r = 0.93$ $p = 1$

Even with all text detected correctly, the recall is not 100%. This depends on the size of the boxes around each text in the ground truth.

Analyzing all these artifacts is helpful to better understand how the methods are evaluated. Note that because of granularity difference, our algorithm performance is sometimes underestimated.

8 Conclusion

We present in this article a complete scheme to process text detection in street level images. In such images, text areas are varying a lot. Focusing on street level in urban context, we develop a text box localization system with as few as possible hypotheses on text shape, size, and style. Looking first for characters, we group them to generate text box candidates that are finally filtered using a texture-based classifier.

We detail every step of the complete process. This work on text detection leads us to introduce an efficient binarization method (TMMS) and to address learning issues. Especially, we efficiently combine a connected component approach and a texture-based approach by cascading them.

We also report a lot of experimentations to validate the approach. Our segmentation process is a full version of the one ranked second in the last DIBCO challenge of ICDAR in 2009. The whole scheme is successfully evaluated on ICDAR database and applied on a real street

level application. The presented scheme is powerful and is now used in ITOWNS project.

This system is definitively adaptable for many applications and contexts. We are currently working on interactive text detection [17] in order to deal with the trickiest text configurations. We are also interested in developing a new evaluation protocol to correct the limitations of the existing metrics that we deeply discuss in our experiments.

Acknowledgements This work is funded by ANR, itowns project 07-MDCO-007-03 [1,22].

References

1. The french national research agency (anr). URL http://www.agence-nationale-recherche.fr/Intl
2. Arth, C., Limberger, F., Bischof, H.: Real-time license plate recognition on an embedded DSP-platform. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR '07) pp. 1–8 (2007)
3. Beucher, S.: Numerical residues. Image Vision Comput. **25**(4), 405–415 (2007). DOI http://dx.doi.org/10.1016/j.imavis.2006.07.020
4. Breen, E.J., Jones, R.: Attribute openings, thinnings, and granulometries. Computer Vision and Image Understanding **64**(3), 377–389 (1996)
5. Chehdi K.; Coquin, D.: Binarisation d'images par seuillage local optimal maximisant un critre d'homognite. GRETSI (1991)
6. Chen, D., Odobez, J., Thiran, J.: A localization/verification scheme for finding text in images and video frames based on contrast independent features and machine learning method. Image Communication **19**(3), 205–217 (2004)
7. Chen, X., Yuille, A.L.: Detecting and reading text in natural scenes. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on **2**, 366–373 (2004). DOI http://doi.ieeecomputersociety.org/10.1109/CVPR.2004.77
8. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning **20**(3), 273–297 (1995)
9. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE CVPR, pp. 886–893. IEEE Computer Society (2005)
10. Ezaki, N., Bulacu, M., Schomaker, L.: Text detection from natural scene images: Towards a system for visually impaired persons. In: 17th International Conference on Pattern Recognition, vol. 2, pp. 683–686 (2004)
11. Fabrizio, J., Cord, M., Marcotegui, B.: Text extraction from street level images isprs workshop cmrt. ISPRS Workshop (2009)
12. Fabrizio, J., Marcotegui, B.: Fast implementation of the ultimate opening. International Symposium on Mathematical Morphology pp. 272–281 (2009)
13. Fabrizio, J., Marcotegui, B., Cord, M.: Text segmentation in natural scenes using toggle-mapping. 2009 IEEE International Conference on Image Processing (2009)
14. Garcia, W.C., Apostolidis, X.: Text detection and segmentation in complex color images. In: IEEE Int. Conf. Acoust., Speech, Signal Process., pp. 2326–2329. IEEE Computer Society (2000)
15. Gatos, B., Ntirogiannis, K., Pratikakis, I.: Icdar 2009 document image binarization contest (dibco 2009). International Conference on Document Analysis and Recognition (2009)
16. Gatos, B., Ntirogiannis, K., Pratikakis, I.: DIBCO 2009: document image binarization contest. International Journal on Document Analysis and Recognition (2010). DOI 10.1007/s10032-010-0115-7
17. Gosselin, P., Cord, M.: Active learning methods for interactive image retrieval. IEEE Transactions on Image Processing **17**(7), 1200–1211 (2008)
18. Hanif, S.M., Prevost, L.: Texture based text detection in natural scene images - a help to blind and visually impaired persons. In: Conference on Assistive Technologies for People with Vision and Hearing Impairments (2007)
19. ICDAR: Robust reading and locating database (2003). URL http://algoval.essex.ac.uk/icdar/TextLocating.html
20. Institut géographique national (ign). URL www.ign.fr
21. Imageval (2006). URL www.imageval.org
22. Anr itowns project. URL www.itowns.fr
23. Joachims: svmlight. URL http://svmlight.joachims.org/
24. Joachims, T.: Making large-scale svm learning practical. Advances in kernel methods: support vector learning pp. 169–184 (1999)
25. Jung, C., Liu, Q., Kim, J.: A stroke filter and its application to text localization. Pattern Recogn. Lett. **30**(2), 114–122 (2009). DOI http://dx.doi.org/10.1016/j.patrec.2008.05.014
26. Jung, K., Kim, K., Jain, A.: Text information extraction in images and video: a survey. Pattern Recognition **37**(5), 977–997 (2004)
27. Kavallieratou, E., Balcan, D., Popa, M., Fakotakis, N.: Handwritten text localization in skewed documents. In: International Conference on Image Processing, pp. I: 1102–1105 (2001)
28. Kuncheva, L.: Combining Pattern Classifiers. Methods and Algorithms. Wiley (2004)
29. Jian Liang, David Doermann, Huiping Li: Camera-Based Analysis of Text and Documents: A Survey. International Journal on Document Analysis and Recognition **7**(2+3), 83 – 104 (2005)
30. Lienhart, R., Effelsberg, W.: Automatic text segmentation and text recognition for video indexing. Multimedia Syst. **8**(1), 69–81 (2000). DOI http://dx.doi.org/10.1007/s005300050006
31. Liu, Q., Jung, C., Kim, S., Moon, Y., yeun Kim, J.: Stroke filter for text localization in video images. IEEE International Conference on Image Processing (2006)
32. Liu, X., Samarabandu, J.: Multiscale edge based text extraction from complex images. In: Int Conf Multimedia Expo, pp. 1721–1724 (2006)
33. Lucas, S.: Icdar 2005 text locating competition results. Eight International Conference on Document Analysis and Recognition (2005)
34. Mancas-Thillou, C.: Natural scene text understanding. Ph.D. thesis, TCTS Lab of the Facult Polytechnique de Mons, Belgium (2006)
35. Niblack, W.: An Introduction to Image Processing. Prentice-Hall, Englewood Cliffs, NJ (1986)
36. Otsu, N.: A threshold selection method from gray level histogram. IEEE Transactions in Systems, Man,. and Cybernetics **9**, 62–66 (1979)
37. Palumbo, P.W., Srihari, S.N., Soh, J., Sridhar, R., Demjanenko, V.: Postal address block location in real time. Computer **25**(7), 34–42 (1992). DOI http://doi.ieeecomputersociety.org/10.1109/2.144438

38. Pan, W., Bui, T.D., Suen, C.Y.: Text detection from natural scene images using topographic maps and sparse representations. In: IEEE ICIP. IEEE Computer Society (2009)
39. Pazio, M., Niedwiecki, M., Kowalik, R., Lebied, J.: Text detection system for the blind. 15th European Signal Processing Conference EUSIPCO pp. 272–276 (2007)
40. Retornaz, T.: Détection de textes enfouis dans des bases d'images généralistes. un descripteur sémantique pour l'indexation. Ph.D. thesis, Ecole Nationale Suprieure des Mines de Paris - C.M.M., Fontainebleau - France (2007)
41. Retornaz, T., Marcotegui, B.: Scene text localization based on the ultimate opening. International Symposium on Mathematical Morphology **1**, 177–188 (2007)
42. Sauvola, J., Inen, M.P.: Adaptive document image binarization. Pattern Recognition **33**, 225–236 (2000)
43. Sauvola, J.J., Seppänen, T., Haapakoski, S., Pietikäinen, M.: Adaptive document binarization. In: ICDAR '97: Proceedings of the 4th International Conference on Document Analysis and Recognition, pp. 147–152. IEEE Computer Society, Washington, DC, USA (1997)
44. Seeger, M., Dance, C.: Binarising camera images for ocr. Proceeding of Sixth International Conference on Document Analysis and Recognition (ICDAR) (2001)
45. Serra, J.: Toggle mappings. From pixels to features pp. 61–72 (1989). J.C. Simon (ed.), North-Holland, Elsevier
46. Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic Imaging **13(1)**, 146–165 (2004)
47. Shafait, F., Keysers, D., Breuel, T.M.: Efficient implementation of local adaptive thresholding techniques using integral images. In: Document Recognition and Retrieval XV. San Jose, CA (2008)
48. Szumilas, L.: Scale and rotation invariant shape matching. Ph.D. thesis, Technische universität wien fakultät für informatik (2008)
49. Trier, O.D., Jain, A.K., Taxt, T.: Feature extraction methods for character recognition-a survey. Pattern Recognition **29**(4), 641 – 662 (1996). DOI DOI: 10.1016/0031-3203(95)00118-2
50. Viola, P., Jones, M.: Robust real-time object detection. In: International Journal of Computer Vision (2001)
51. Wahl, F., Wong, K., Casey, R.: Block segmentation and text extraction in mixed text/image documents. Computer Graphics and Image Processing **20**(4), 375–390 (1982)
52. Wolf, C., michel Jolion, J., Chassaing, F.: Text localization, enhancement and binarization in multimedia documents. In: In Proceedings of the International Conference on Pattern Recognition (ICPR) 2002, pp. 1037–1040 (2002)
53. Wolf, C., Jolion, J.M.: Object count/area graphs for the evaluation of object detection and segmentation algorithms. IJDAR **8**(4), 280–296 (2006)
54. Xiao, Y., Yan, H.: Text region extraction in a document image based on the delaunay tessellation. Pattern Recognition **36**(3), 799–809 (2003)
55. Zhao, X., K. Lin, Y.F., Hu, Y., Y. Liu, T.H.: Text from corners: A novel approach to detect text and caption in videos. IEEE Transactions on Image Processing **20**(3), 790–799 (2011)
56. Zhu, K., F. Qi, R.J., Xu, L., Kimachi, M., Wu, Y., Aziwa, T.: Using adaboost to detect and segment characters from natural scenes. In Proc. of CBDAR, ICDAR Workshop (2005)